TUTORIAL DASAR

MEMBUAT PROJECT ARDUINO UNO

Pendahuluan

Arduino merupakan suatu perangkat yang dirancang dengan kemampuan komputasi yang dapat berinteraksi secara lebih dekat dengan dunia nyata dibandingkan komputer biasa. Satu hal yang perlu dicatat ialah Arduino bersifat open source, baik microcontroller board yang dapat dimodifikasi, begitu juga dengan development environment untuk menuliskan source code program yang disebut dengan istilah sketch.

Sejarah Singkat
Arduino bermula pada tahun 2005, sebagai sebuah project bagi para pelajar di Interaction Design Institute Ivrea di Ivrea, Italia. Pada saat itu para pelajar menggubakan BASIC Stamp yang cukup mahal bagi pelajar. Adalah Massimo Banzi, salah satu pendiri Arduino, yang sekaligus mengajar di Ivrea, mencoba menyelesaikan persoalan tersebut. Asal mula nama Arduino berasal dari nama sebuah bar yang ada di Ivrea, dimana para pendiri Arduino biasa berkumpul disana. Untuk hardware wiring awal dirancang oleh Hernando Barragan. Bersama dengan David Cuartielles, mulai memperkenalkan Arduino.

Feature Kelebihan
Arduino dapat digunakan untuk mengembangkan suatu sistem interaktif, yang menerima input dari bermacam switch atau sensor, dan mengendalikan bermacam hal semisal lampu, motor, dan output lainnya.
Ada berbagai macam jenis microcontroller dan platform yang tersedia untuk keperluan komputasi fisik. Arduino dirancang secara sederhana dan mudah dipelajari, dikembangkan untuk keperluan project tugas sekolah, kuliah, ataupun tugas akhir. Adapun kelebihan yang coba ditawarkan Arduino antara lain.

- Relatif murah
- Bersifat cross-platform dapat dijalankan di berbagai operating system seperti Windows, Macintosh OSX, dan Linux
- Sederhana, dengan programming environment turunan dari bahasa pemrograman C yang mudah dimengerti
- Baik hardware maupun software bersifat open source

Untuk dasar tutorial pemrograman Arduino berikut ini beberapa function yang biasa digunakan.

setup()
Digunakan untuk inisialisasi variable, pin mode, penggunaan library, dan lain sebagainya. Hanya dijalankan sekali, pada saat Arduino pertama kali dinyalakan, atau setelah reset.

loop()
Setelah function setup(), digunakan function loop() yang sesuai dengan namanya, untuk

menjalankan program utama dalam Arduino secara berulang terus-menerus, hingga Arduino dimatikan atau reset.

pinMode()
Digunakan untuk melakukan konfigurasi secara spesifik fungsi dari sebuah pin, apakah digunakan sebagai input atau sebagai output. Contoh penggunaan function pinMode() ialah sebagai berikut.

- **pinMode(0, INPUT)** konfigurasi pin 0 Arduino sebagai pin input
- **pinMode(13, OUTPUT)** konfigurasi pin 13 Arduino sebagai pin output

digitalRead()
Digunakan untuk membaca nilai pin digital yang spesifik, apakah bernilai HIGH atau LOW. Contoh penggunaan function digitalRead() ialah sebagai berikut.

- **digitalRead(0)** membaca nilai digital dari pin 0 Arduino

digitalWrite()
Selain membaca nilai, ada juga function untuk menuliskan atau memberikan nilai pada suatu pin digital secara spesifik. Dengan function digitalWrite() memberikan nilai pin digital yang spesifik apakah bernilai HIGH atau LOW, dapat dilakukan. Contoh penggunaan function digitalWrite() ialah sebagai berikut.

- **digitalWrite(13, HIGH)** memberikan nilai digital HIGH pada pin 13 Arduino

delay()
Sesuai dengan namanya, function delay() digunakan untuk memberikan waktu tundaan (dalam satuan millisecond) untuk mengerjakan satu baris program ke baris selanjutnya. Contoh penggunaan function delay() ialah sebagai berikut.

- **delay(1000)** memberikan waktu tundaan 1000 millisecond, atau setara dengan 1 detik sebelum melanjutkan mengerjakan perintah baris program selanjutnya

analogRead()
Selain membaca nilai digital, Arduino juga dapat digunakan untuk membaca nilai analog. Dengan menggunakan function analogRead(), untuk membaca nilai analog melalui pin analog. Untuk board Arduino UNO memiliki 6 channel analog, Arduino Mini dan Nano 8 channel, sedangkan Arduino Mega 10 channel, dengan resolusi 10 bit analog to digital converter. Dengan resolusi 10 bit memungkinkan pemetaan tegangan antara 0 volt hingga 5 volt dalam nilai integer dari 0 hingga 1023. Sehingga resolusi pembacaan nilai analog ialah 5 volt dibagi 1024 unit, atau sekitar 4,9 mV per unit. Dibutuhkan sekitar 100 microsecond untuk membaca suatu input analog, dengan kata lain tingkat pembacaan maximum nilai analog ialah 10000 kali dalam satu detik.

analogReference(type)
Digunakan untuk memberikan nilai tegangan referensi untuk input analog, dengan pilihan type yang ada antara lain.

- **DEFAULT**, nilai tegangan default untuk board Arduino ialah 5 volt atau 3,3 volt
- **INTERNAL**, sebuah referensi built-in, setara tegangan 1,1 volt pada ATmega168 atau ATMega 328, dan 2,56 volt untuk ATMega8
- **INTERNAL1V1**, sebuah referensi built-in, tegangan 1,1 volt, untuk Arduino Mega
- **INTERNAL2V56**, sebuah referensi built-in, tegangan 2,56 volt, untuk Arduino Mega
- **EXTERNAL**, sebuah referensi referensi dari pin AREF, nilai tegangan berkisar anatar 0 volt hingga 5 volt

Untuk informasi lebih lanjut silahkan kunjungi situs halaman Arduino.

Arduino Operasi Bitwise

Bahasa C yang digunakan pada Arduino, menjadikannya mudah untuk dipelajari oleh pelajar, mahasiswa, dari mulai untuk project sederhana hingga tugas akhir kuliah. Bahasa C menawarkan operasi manipulasi bit, yang mana pada Arduino bisa sangat berguna untuk mengubah suatu nilai bit dalam data yang berupa byte. Implementasi nyata pada Arduino ialah pada manipulasi nilai pin-pin Arduino berubah dari kondisi high menjadi low, atau sebaliknya. Dalam hal ini operasi manipulasi bit dimungkinkan dengan penerapan gerbang logika atau operasi fungsi logika yang disebut bitwise. Untuk operasi bitwise standard yang biasa digunakan ialah & (AND), | (OR), << (Left Shift) menggeser nilai bit ke kiri, dan >> (Right Shift) menggeser nilai bit ke kanan.

AND
Fungsi operasi logika AND dari dua buah operand akan menghasilkan output dengan logika 1, high, hanya pada saat kedua bit pada tingkat yang sama juga bernilai logika 1, high. Selain itu akan menghasilkan output dengan logika 0, low. Berikut ini adalah contoh potongan sktech Arduino.

```
byte operand1 = B00001111;
byte operand2 = B01010101;
byte result = operand1 & operand2;
```

Penjelasannya sebagai berikut.

- Nilai operand1 dengan tipe data byte ialah 00001111
- Nilai operand2 dengan tipe data byte ialah 01010101
- Nilai result dengan tipe data byte ialah hasil operasi AND byte dari operand1 dengan operand2

```
AND

operand1 = 00001111
operand2 = 01010101
_____________________
result   = 00000101
```

OR

Fungsi operasi logika OR dari dua buah operand akan menghasilkan output dengan logika 1, high, hanya pada saat salah satu atau kedua bit pada tingkat yang sama bernilai logika 1, high. Selain bila kedua bit pada tingkat yang sama bernilai logika 0, low maka akan menghasilkan output dengan logika 0, low. Berikut ini adalah contoh potongan sktech Arduino.

```
byte operand1 = B00001111;
byte operand2 = B01010101;
byte result = operand1 | operand2;
```

Penjelasannya sebagai berikut.

- Nilai operand1 dengan tipe data byte ialah 00001111
- Nilai operand2 dengan tipe data byte ialah 01010101
- Nilai result dengan tipe data byte ialah hasil operasi OR byte dari operand1 dengan operand2

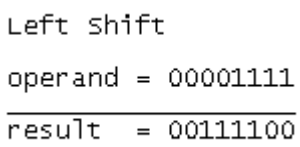
```
OR

operand1 = 00001111
operand2 = 01010101
_____________________
result   = 01011111
```

<< Left Shift
Fungsi dari bitwise << ialah menggeser nilai-nilai bit ke kiri. Menggeser nilai-nilai bit ke kiri, akan menghasilkan output yang bernilai dua kali lipat dari operand awalnya. Berikut ini adalah contoh potongan sktech Arduino.

```
int i = 2;
byte operand = B00001111;
byte result = operand << i;
```

Penjelasannya sebagai berikut.

- Nilai operand dengan tipe data byte ialah 00001111
- Nilai result dengan tipe data byte ialah hasil pergeseran bit dari operand ke kiri sebanyak i, yakni 2 kali

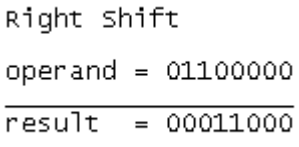


>> Right Shift
Fungsi dari bitwise >> ialah menggeser nilai-nilai bit ke kanan. Menggeser nilai-nilai bit ke kanan, akan menghasilkan output yang bernilai separuh dari operand awalnya. Berikut ini adalah contoh potongan sktech Arduino.

```
int i = 2;
byte operand = B01100000;
byte result = operand >> i;
```

Penjelasannya sebagai berikut.

- Nilai operand dengan tipe data byte ialah 01100000
- Nilai result dengan tipe data byte ialah hasil pergeseran bit dari operand ke kanan sebanyak i, yakni 2 kali



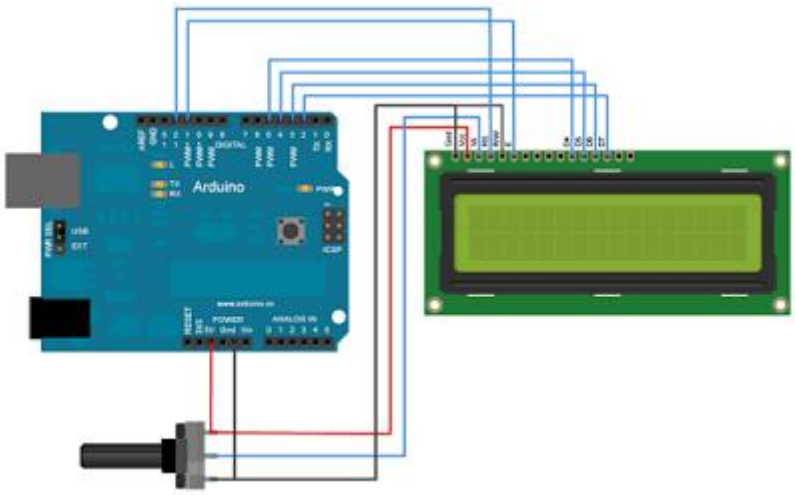
Lihat juga mengenai implementasinya dalam program Arduino LED bitwise operation.

Arduino Serial

Setiap board Arduino setidaknya telah memiliki satu buah serial port, yang memungkinkannya untuk melakukan proses pertukaran data dengan komputer atau perangkat lain melalui jalur komunikasi serial. Pada Arduino Uno, dengan menggunakan RX (pin 0) dan TX (pin 1) memungkinkan proses komunikasi serial tersebut. Oleh karena itu, pin 0 dan pin 1, yang sedang digunakan untuk proses pertukaran data komunikasi serial, tersebut tidak dapat digunakan sebagai pin input atau juga output.
Arduino IDE sendiri telah mencakup feature serial monitor untuk berkomunikasi dengan board Arduino. Satu hal yang perlu dicatat ialah menyesuaikan baud rate antara Arduino dengan komputer, agar proses pertukaran data antara keduanya dapat berjalan dengan baik. Agar lebih jelas mengenai baud rate, berikut penjelasannya.
Pada telekomunikasi dan elektronika, baud (dalam Bd) merupakan unit untuk merepresentasikan banyaknya modulasi atau pulse setiap satuan waktu. Dalam sistem digital, (yang menggunakan nilai discrete) dengan binary code, 1 Bd sama dengan 1 bit per second. Untuk nama unit baud ini sendiri, berasal dari Emile Baud, yang merupakan penemu dari Baudot code untuk telegrapy.
Dalam proses pertukaran data antara microcontroller dengan komputer, biasa melalui komunikasi serial. Disini kaitannya dengan baud rate, seberapa cepat data dikirimkan melalui jalur komunikasi serial tersebut. Agar dua buah perangkat, microcontroller dengan komputer dapat bertukar data dengan baik, maka baud rate antara keduanya harus setara. Biasa baud rate yang digunakan ialah 9600 bits per second. Adapun baud rate lainnya bisa bernilai 1200, 2400, 4800, 19200, 38400, 57600, dan 115200. Setiap nilai baud rate tersebut memiliki ratio perbandingan kelipatan dari baud rate standard 9600 yang biasa digunakan.
Semakin tinggi baud rate, maka semakin cepat data dikirimkan, demikian pula sebaliknya. Meski begitu tetap ada batasan nilai baud rate, bila melebihi nilai batasan tersebut maka akan terjadi error.
Lihat juga mengenai interface komunikasi serial.

Arduino LCD

Salah satu jenis penampil yang sering digunakan ialah LCD (Liquid Crystal Display), yang memiliki banyak kelebihan dibandingkan seven segment display. Salah satu kelebihan LCD ialah menampilkan pesan text. Jenis LCD yang banyak digunakan untuk project microcontroller sederhana ialah LCD 16x2, 16 kolom dengan 2 baris.
Untuk memulai belajar menggunakan penampil LCD, terlebih dahulu dilakukan pemetaan konfigurasi wiring diagram. Hal ini ditujukan agar pin yang digunakan pada source code dan wiring diagram sesuai. Berikut ini ialah wiring diagram yang digunakan antara Arduino dengan LCD pada uji coba awal.



Berdasarkan gambar wiring diagram tersebut.

- LCD RS pin dihubungkan ke digital pin 12
- LCD Enable pin dihubungkan ke digital pin 11
- LCD D4 pin dihubungkan ke digital pin 5
- LCD D5 pin dihubungkan ke digital pin 4
- LCD D6 pin dihubungkan ke digital pin 3
- LCD D7 pin dihubungkan ke digital pin 2

Penggunaan potentiometer (variable resistor), ditujukan untuk mengatur tingkat gelap-terang tampilan LCD tersebut, berdasarkan fungsi pembagi tegangan. Berikut ini ialah source code (sketch) sederhana untuk menampilkan pesan text.

```
/*
Program LCD sederhana
dengan mencakup library LCD
Loki Lang
*/

#include <LiquidCrystal.h>

LiquidCrystal lang(12, 11, 5, 4, 3, 2);

void setup()
{
  lang.begin(16, 2);
}

void loop()
{
  lang.clear();
  lang.setCursor(3, 1);
  lang.print("Manchester");
  lang.setCursor(5, 1);
  lang.print("United");
}
```

Dalam source code tersebut digunakan macro #include preprocessed directive yang memuat file header LiquidCrystal.h untuk library LCD, untuk penampil LCD.

Proses inisialisasi pin Arduino yang terhubung ke pin LCD RS, Enable, D4, D5, D6, dan D7, dilakukan dalam baris LiquidCrystal lang(12, 11, 5, 4, 3, 2); dimana lang merupakan variable yang dipanggil setiap kali instruksi terkait LCD akan digunakan. Catatan, untuk nama variable lang dapat diubah dengan lcd atau nama lainnya, selama dalam pemanggilan namanya sesuai.

begin()
Untuk begin() digunakan dalam inisialisasi interface ke LCD dan mendefinisikan ukuran kolom dan baris LCD. Pemanggilan begin() harus dilakukan terlebih dahulu sebelum memanggil instruksi lain dalam library LCD. Untuk syntax penulisan instruksi begin() ialah sebagai berikut.
lang.begin(cols, rows)
Dengan lang ialah nama variable, cols jumlah kolom LCD, dan rows jumlah baris LCD.

clear()
Instruksi clear() digunakan untuk membersihkan pesan text. Sehingga tidak ada tulisan yang ditapilkan pada LCD.

setCursor()
Instruksi ini digunakan untuk memposisikan cursor awal pesan text di LCD. Penulisan syntax setCursor() ialah sebagai berikut.
lang.setCursor(col, row)
Dengan lang ialah nama variable, col kolom LCD, dan row baris LCD.

print()
Sesuai dengan namanya, instruksi print() ini digunakan untuk mencetak, menampilkan pesan text di LCD. Penulisan syntax print() ialah sebagai berikut.
lang.print(data)
Dengan lang ialah nama variable, data ialah pesan yang ingin ditampilkan.

Lihat juga mengenai tutorial dan penjelasan mengenai project Arduino Uno LED blinking dan Arduino Uno LED animation.

Arduino Timer

Pada ulasan belajar Arduino kali ini membahas tentang timer dan interrupt. Banyak function dalam Arduino menggunakan timer semisal delay(), delayMicroseconds(), millis(), dan micros() untuk penggunaannya masing-masing akan dijelaskan sebagai berikut.

- delay(), digunakan untuk tundaan eksekusi baris program selanjutnya dalam millisecond
- delayMicroseconds(), digunakan untuk tundaan eksekusi baris program selanjutnya dalam microseconds
- millis(), digunakan sebagai pewaktu internal yang (bila tanpa terminate bersyarat) akan terus berjalan hingga terjadi overflow (kembali ke nilai 0) dengan unit dalam millisecond, untuk board Arduino Uno nilai millis() akan terus berjalan hingga sekitar 50 hari
- micros(), digunakan sebagai pewaktu internal yang (bila tanpa terminate bersyarat) akan terus berjalan hingga terjadi overflow (kembali ke nilai 0) dengan unit dalam microsecond, untuk board Arduino Uno nilai millis() akan terus berjalan hingga sekitar 70 jam

Sebuah pewaktu, timer, merupakan bagian dari microcontroller yang berperan sebagai clock internal untuk mengukur waktu suatu event. Untuk timer dapat diatur dengan menggunakan beberapa register khusus. Pada firmware Arduino semua timer memiliki konfigurasi frekuensi 1 kHz dengan enable interrupt. Berikut ini timer khusus untuk Arduino.

- Timer0, 8 bit, digunakan untuk function seperti delay(), millis(), dan micros(), dengan mengubah konfigurasi Timer0 akan mempengaruhi function lainnya
- Timer1, 16 bit, biasa digunakan untuk aplikasi terkait motor servo
- Timer2, 8 bit, function tone() menggunakan Timer2

Timer Register

Untuk dapat melakukan manipulasi timer pada Arduino terlebih dahulu harus mengetahui fungsi dari masing-masing register yang terkait timer. Salah satu register timer yang paling penting ialah TCCRx (Timer/Conter Control Register), dengan x adalah nomor, berikut ini adalah register apa saja yang digunakan untuk timer.

- TCCRx (Timer/Conter Control Register), dimana prescaler dapat dikonfigurasi disini sekaligus mode operasi timer
- TCNTx (Timer/Counter Register), dimana nilai timer disimpan, merupakan register pencacah mulai dari 0 hingga nilai maximum
- OCRx (Output Compare Register), untuk membandingkan OCR yag diberikan dengan nilai TCNT
- ICRx (Input Capture Register), hanya tersedia untuk timer 16 bit, menerima data timer
- TIMSKx (Timer/Counter Interrupt Mask Register), digunakan untuk menjalankan atau mematikan timer interrupt
- TIFRx (Timer/Counter Interrupt Flag Register), menandakan timer interrupt hasil operasi timer